Adaptive controlled Mode-locked Fiber Laser

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fullfillment of the Requirements for the Degree of

Master of Science

By

Yi Gu

August 2014

Acknowledgements

First, I would like to thank Prof. Wise for his support as my professor. I am very grateful to him for finding a really interesting and promising project for me. Also, Frank helped me a lot, not only in problems in the project, but also the methods to work with challenging problems.

I would also like to thank William Renninger, Erin Stranford, Liu Hui, Logan Wright, Zhanwei Liu, Yuxing Tang and Xiaosheng Xiao. I am grateful to them for their help and kindness through my project. I am very grateful to Zhanwei Liu for the stimulating discussion in ESC Control.

Finally, I would like to thank my parents, my family, my girlfriend and all my friends. Without their support, I would not finish my project.

Contents

1. Abstract

In this thesis, an algorithm termed Extremum-seeking control (ESC) is employed to adaptively control an all-normal-dispersion (ANDi) fiber laser. ESC adaptively controls the angle of a half waveplate which is a key component in the laser to track the optimized performance and stables the laser. In this thesis, ANDi laser and ESC are introduced. Then experiment results is presented and discussed.

2. Introduction

Ultrashort pulsing technology has huge impact both commercially and scientifically. Mode-locked lasers are essential resources for ultrashort pulses. Mode-locked lasers generate ultrashort pulses by locking multiple axial modes in the laser cavity. Since the first Ti:sapphire mode-locked laser was developed in 1990's, ultrashort pulsing technology has been widely applied in industry and scientific research, such as multi-photon imaging, frequency metrology, thin film characterization, etc.

In recent decades, fiber based mode-locked laser are gaining more concern because they are smaller and less expensive than their solid state counterparts. Although the performance of fiber based mode-locked laser is not comparable to solid state lasers. It's advantages has made it successful in applications such as endoscopy and precision material processing.

The specific fiber based mode-locked laser investigated in this thesis is all normal dispersion (ANDi) fiber laser[1]. ANDi fiber laser is compact, cheap and reliable. However, randomness of fiber sets and variation of parameters in the cavity caused by environmental variation requires development of methods to stabilize the laser. Also, enhancing the performance of ANDi laser by optimizing the parameters in the cavity is an interesting subject. One important method to make ANDi laser stable and enhance its performance is to manipulate the nonlinear polarized rotation (NPR)[2,3], which is an essential part in the cavity. To adaptively control the NPR in the cavity, we apply a particular algorithm termed extremum-seeking control algorithm (ESC)[4]. Extremum-seeking control (ESC) is an adaptive method of finding local maxima of an objective function on the output of a dynamical system. By applying ESC on ANDi laser, we stabled the laser as well as enhanced its performance.

This thesis is arranged as follows: First, we introduce ANDi laser. Second, we introduce ESC. Third, we introduced our experiment setup. Forth, we show our experiment results and duscuss our results. The code and software environment is introduced in appendix.

3. ANDi laser

ANDi laser is all normal dispersion mode-locked fiber laser with compact and simple cavity structure. Because of its all normal dispersion, pulses in the laser have large linear chirp. The chirped pulse enables the laser to generate stable pulses with an order of magnitude higher pulse energy than in a soliton laser.

(1) Cavity structure

The cavity structure of ANDi laser is shown in Figure.1.

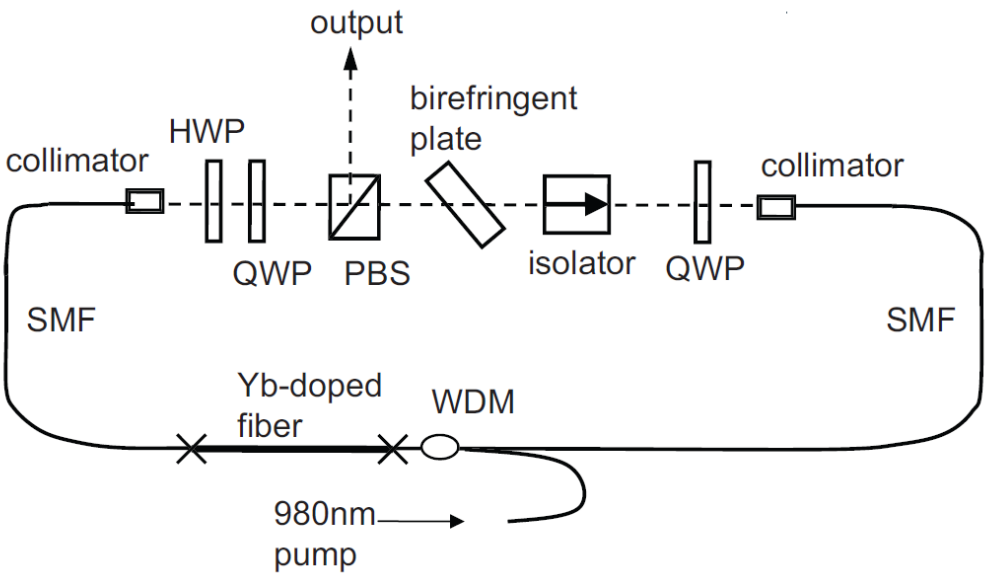

Fig.1 Cavity scheme of ANDi laser

Where HWP is half waveplate, PBS is polarized beam splitter, QWP is quarter waveplate, SMF is single mode fiber, WDM is wavelength division multiplexer. This is the exact laser we use in our experiment. The specific parameter is as follows: We use YB-doped fiber as gain fiber and a 980 single mode diode as pump source. The laser operates at 1030nm. This is a ring cavity and the operation direction is shown by the isolator. The birefringence plate works as a 12nm filter. In this specific cavity, we have 3m SMF before the gain fiber and 1m SMF after the gain fiber.

(2) Pulse-shaping mechanism

(i) Qualitative pulse-shaping mechanism

There are three main elements in the pulse-shaping mechanism[5]. The linear chirp, the nonlinearity and the wavelength filter. Pulses pick up linear chirp in the whole cavity equally. Because nonlinearity is proportion to the intensity, pulses pick up some nonlinear phase before the gain fiber and most nonlinear phase after the gain fiber. Both linear chirp and nonlinearity make pulses spread in time domain. Pulses are cut by wavelength filter. Because pulses in the cavity are highly chirped, the pulses have their high frequency part proceeding low frequency part. By cutting both high and low frequency part of the pulses, pulses are compressed in time domain. NPR is an essential component in the cavity. The transmission rate of NPR is proportion to the intensity. Field with higher intensity has larger transmission rate. NPR starts pulse formation from noise.

Three aspects determines the pulse quality, they are noise or input pulses, cavity structure and NPR. In this thesis, we monitor the NPR to achieve pulses with larger pulse energy and shorter pulse duration. The pulse-shaping mechanism is shown in Figure.2 and Figure.3.

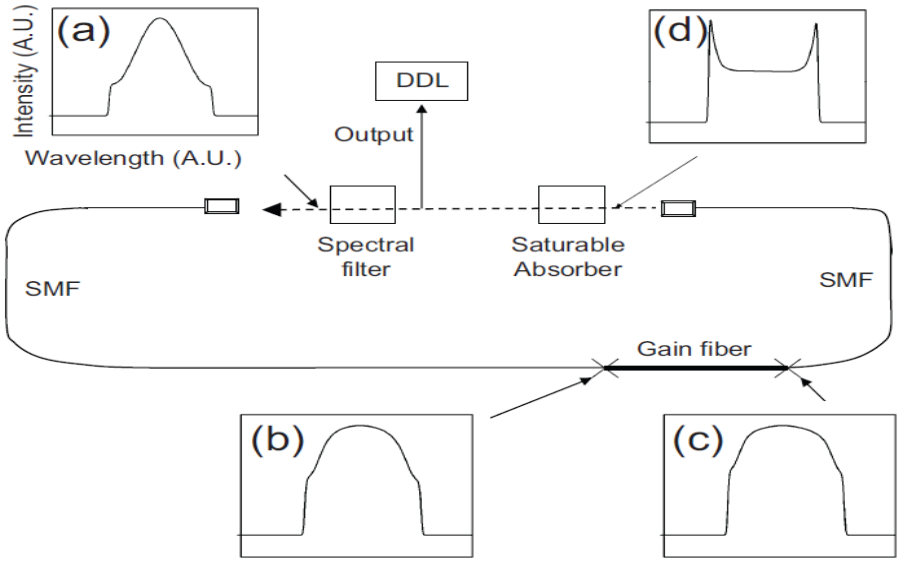

Fig.2 pulses at different positions of the cavity

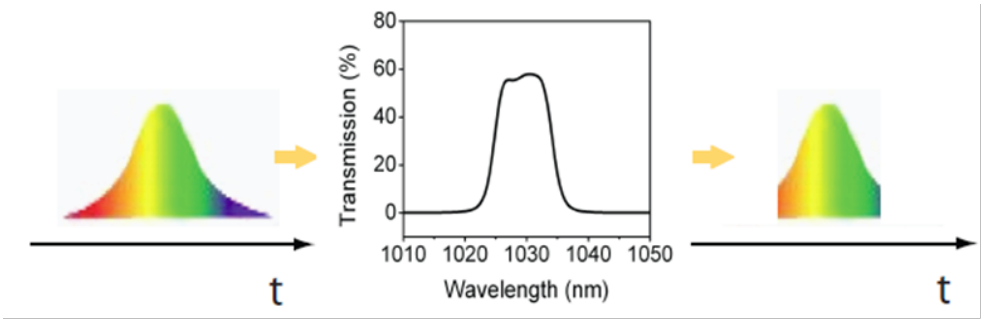

Fig.3 Pulse compression process in the cavity

(ii) Quantitative analyze of pulse-shaping mechanism

We use cubic-quintic Ginzberg-Landau Equation (CQGLE) to analyze intra-cavity pulse evolution.

$$U_z = gU + (\frac{1}{\Omega} - i\frac{D}{2})U_{tt} + (\alpha + i\gamma)|U|^2 U + \delta|U|^4 U$$

Where D is group velocity dispersion which leads to the linear chirp, γ is nonlinearity, g is gain in the cavity, α and δ describe the NPR and $\frac{1}{\Omega}$ describes the Gaussian wavelength filter. Gain saturation is modeled as:

$$g = \frac{g_0}{1 + E_{pulse} / \mathrm{E}_{sat}}$$

D, γ, g are determined by the fiber in the cavity. $\frac{1}{\Omega}$ is determined by the wavelength filter. One important thing to mention, α and δ terms are approximation of the NPR which is not accurate. The experimental NPR may well be modeled by terms above quantic. Also, NPR will vary as temperature changing. So we cannot get accurate solution from CQGLE. We need ESC algorithm to stable and optimize our solution.

(3) Key components in the cavity

(i) Nonlinear Polarized Rotation (NPR)

NPR is an ultra-fast absorber which relies on nonlinearity in the fiber. NPR has higher transmission rate with higher intensity, which starts mode-locking from noise. NPR is a highly tunable component with high performance. However, it's sensitive to environmental perturbations. NPR consists of two

quarter waveplates, one half waveplate, a polarizer and nonlinear fiber. The schematic for NPR is shown in Figure.4. Where $\chi^{(3)}$ is third order nonlinearity of the fiber. α_1, α_2 and α_3 are angles of waveplates. Arrows show the direction of the ring laser.

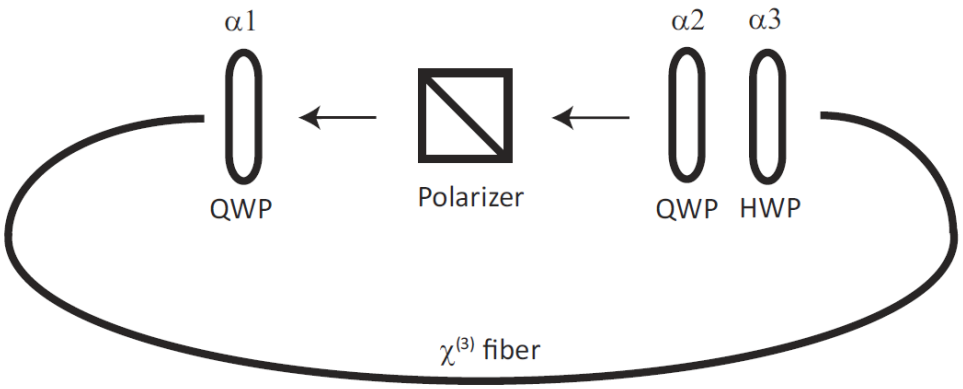

Fig.4 Schematic for NPR

The performance of NPR can be analyzed using Jones matrix. Jones matrix for QWP is:

$$QWP[\phi]=\frac{1-i}{2}\begin{pmatrix} i+\cos(2\phi) & \sin(2\phi) \\ \sin(2\phi) & i-\cos(2\phi) \end{pmatrix}.$$

Jones matrix for HWP is:

$$HWP[\phi]=\begin{pmatrix} \cos(2\phi) & \sin(2\phi) \\ \sin(2\phi) & -\cos(2\phi) \end{pmatrix}.$$

Where ϕ is the rotating angle of the waveplate.

Jones matrix for polarizer is:

$$P=\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Neglecting dispersion and higher order nonlinearity, the couple-mode equation for passive fiber is:

$$\frac{\partial u}{\partial z}=i\gamma(|u|^2+\frac{2}{3}|v|^2)u+\frac{i\gamma}{3}u^*v^2e^{\frac{-4\pi iz}{L_B}},$$
$$\frac{\partial v}{\partial z}=i\gamma(|v|^2+\frac{2}{3}|u|^2)v+\frac{i\gamma}{3}v^*u^2e^{\frac{-4\pi iz}{L_B}}.$$

Where, u and v are two linear polarization basis, z is the propagation length, γ is the nonlinearity, L_B is the beat length.

Because beat length is around 20m, so we can assume beat length is infinite. Thus, the solution of the equation gives Kerr matrix :

$$Kerr = e^{i\gamma(|u(0)|^2+|u(0)|^2)L_{eff}} \begin{pmatrix} \cos(\frac{2}{3}\gamma \operatorname{Im}[u(0)v^*(0)]) & \sin(\frac{2}{3}\gamma \operatorname{Im}[u(0)v^*(0)]) \\ -\sin(\frac{2}{3}\gamma \operatorname{Im}[u(0)v^*(0)]) & \cos(\frac{2}{3}\gamma \operatorname{Im}[u(0)v^*(0)]) \end{pmatrix}.$$

Assuming the initial polarization is parallel to the polarizer and with the equations above, we can calculate the transmission of NPR as follows:

$$\begin{pmatrix} u_{n+1}(\mathrm{t}) \\ 0 \end{pmatrix} = P \cdot QWP[\alpha_2] \cdot HWP[\alpha_3] \cdot Kerr \cdot QWP[\alpha_1] \cdot \begin{pmatrix} u_n(\mathrm{t}) \\ 0 \end{pmatrix}.$$

Thus, the transmission rate of NPR is:

$$T(\alpha_1, \alpha_2, \alpha_3^*, I) = A + B\cos(I\omega) + \mathrm{C}\sin(I\omega).$$

Where,

$$A = \frac{1}{8}(4\sin(2\alpha_1)\sin(2\alpha_2) + 4),$$

$$B = \frac{1}{2}\cos(2\alpha_1)\cos(2\alpha_2)\cos(2\alpha_3^*),$$

$$C = -\frac{1}{2}\cos(2\alpha_1)\cos(2\alpha_2)\sin(2\alpha_3^*),$$

$$\omega = \frac{2}{3}\sin(2\alpha_1)$$

The transmission rate is determined by the angles of waveplates and intensity. So the performance of NPR is highly tunable. By simply rotating the waveplates, transmission rate of NPR can be changed. So we can mode lock the laser and change the mode-locking state by rotating the three waveplates. However, the polarization states in the fiber is not stable, and NPR is sensitive to temperature. Also, different transmission rate of NPR leads to different pulse quality, and it's impossible to optimize the laser by manually rotating the waveplates. Thus, we need an algorithm not only to stable the laser, but also to optimize the performance of NPR.

(ii) Birefringence filter

A birefringence filter is a spectral filter consists of a birefringence plate and a polarizer. The orthogonal polarization states accumulate different wavelength dependent phase shift, which rotates the polarization states. Since phase

shift is wavelength dependent, different wavelength lead to different polarization rotation. Thus, the polarizer will cause a wavelength dependent loss.

The Jones matrix for birefringence plate is as follow, where λ is wavelength, θ is the angle between optical axis of the birefringence plate and the input polarization, d is thickness of the birefringence plate. n_e and n_o are the extraordinary and ordinary refraction index.

$$M(\lambda,\theta,\mathrm{d}) = \begin{pmatrix} e^{\frac{2i\mathrm{d}n_e\pi}{\lambda}}\cos^2(\theta)+e^{\frac{2i\mathrm{d}n_o\pi}{\lambda}}\sin^2(\theta) & \left(e^{\frac{2i\mathrm{d}n_e\pi}{\lambda}}-e^{\frac{2i\mathrm{d}n_o\pi}{\lambda}}\right)\cos(\theta)\sin(\theta) \\ \left(e^{\frac{2i\mathrm{d}n_e\pi}{\lambda}}-e^{\frac{2i\mathrm{d}n_o\pi}{\lambda}}\right)\cos(\theta)\sin(\theta) & e^{\frac{2i\mathrm{d}n_o\pi}{\lambda}}\cos^2(\theta)+e^{\frac{2i\mathrm{d}n_e\pi}{\lambda}}\sin^2(\theta) \end{pmatrix}$$

Thus, the transmission rate of the birefringence filter is:

$$T = \left| e^{\frac{2i\mathrm{d}n_e\pi}{\lambda}}\cos^2(\theta)+e^{\frac{2i\mathrm{d}n_o\pi}{\lambda}}\sin^2(\theta) \right|^2 .$$

Assuming θ is 45°, we can simplify the above expression as:

$$T = \cos^2\left(\frac{\pi d(n_e-n_o)}{\lambda}\right).$$

So by choosing birefringence plates with different thickness, we can get spectral filter with different bandwidth.

4. Extremum-seeking control algorithm

Extremum-seeking control (ESC) is an adaptive control law that finds and tracks local maxima of an objective function by sinusoidally varying a set of input parameters and measuring the consequent variation of the objective function. In ESC, the measured variation in objective function is compared with the input variation to estimate an optimal of input parameters. So we don't need to know the relation between input parameters and objective function, which makes ESC especially useful for controlling the NPR for ANDi laser, because the system is complex and nonlinear.

In this thesis, we only control one HWP in NPR, so we use a single input and a single output (SISO) ESC.

(1) Schematic for SISO ESC

The schematic for SISO ESC is shown in Figure.5.

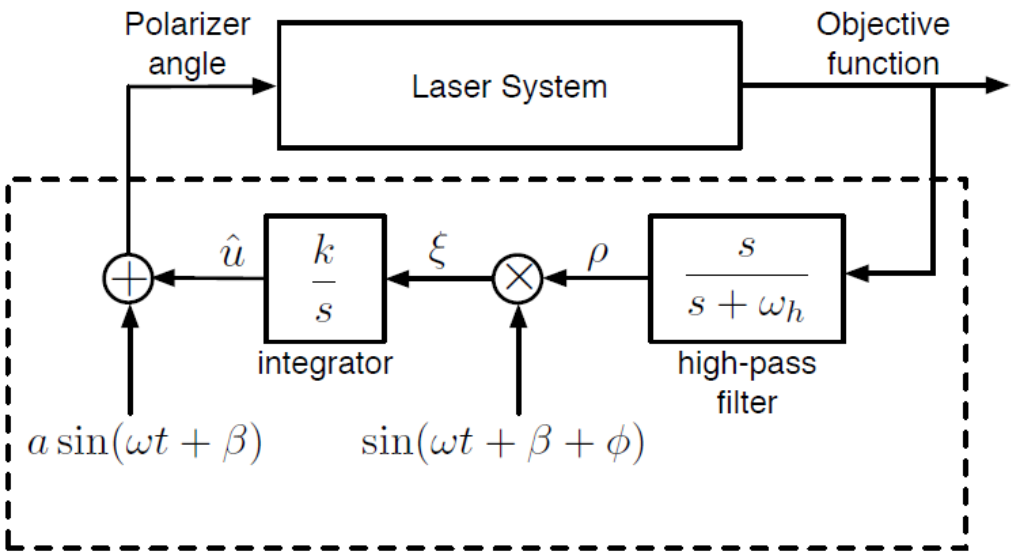

Fig.5 Schematic for SISO ESC

As shown in Figure.5. The angle of HWP is the input. Then the objective function which will be introduced later is measured. And the measured objective function passes a high-pass filter, so as only the variation of objective function goes through. Then the output from high-pass filter is multiplied by a sin signal with the same frequency as the input variation. The phase ϕ is to compensate the phase additional phase shift caused by high-pass filter and integrator to ensure that ESC has a best efficiency. ϕ is obvious to calculate:

$$\phi = \tan^{-1}(\frac{\omega}{\omega_h}) - \frac{\pi}{2}$$

This multiplication is used to compare the variation of input and objective function. Then, the output is multiplied by a constant ξ and passes an integrator. Finally the output from the integrator is added to the input.

(2) Properties of SISO ESC

(i) Mechanism of SISO ESC

How the algorithm works is introduced briefly above. One important thing to clarify is how the algorithm compares the variation of input and output objective function. The mechanism is shown in Figure.6.

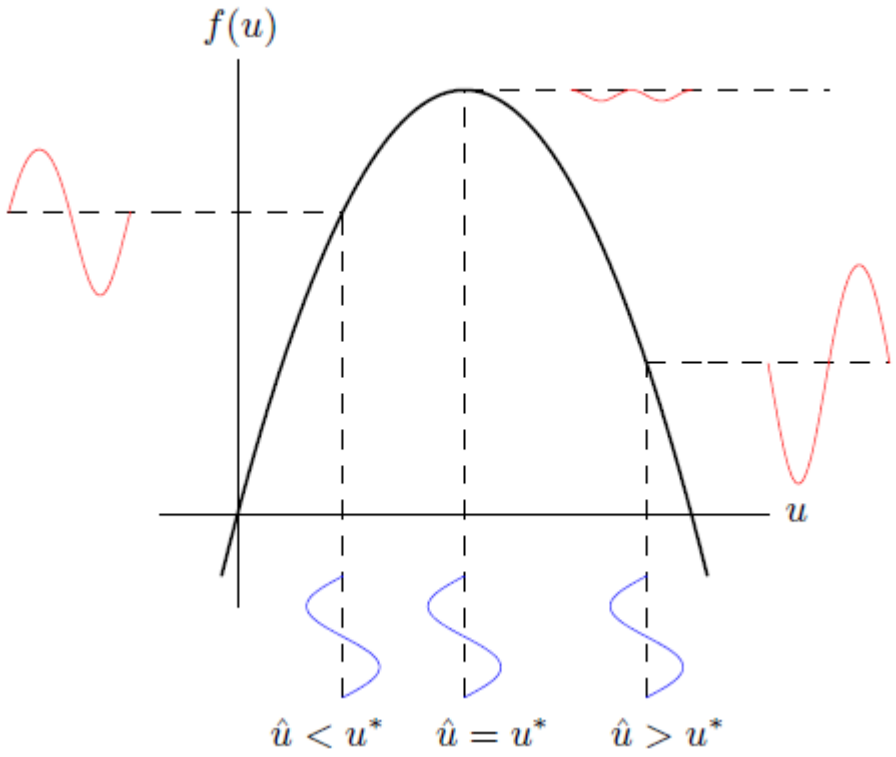

Fig.6 Mechanism of SISO ESC

We assume a relation between input parameter and output objective function. In reality, f(u) can have any form. u is the input sinusoidal variation. The red signal is the variation of output objective function, which is ρ in Figure.5. multiply red variation and blue variation and integrate it in time domain, we get positive value if function f is increasing, negative value if f is decreasing, and when we reach the maximum, the integrating is zero. Thus, SISO ESC tracks local maxima. However, the disadvantage is that SISO ESC is easy to get trapped in local maxima which might be very low. In that situation, it will never reach global maxima. There are several methods to avoid being trapped in low local maxima. We can use a single input multi output ESC to avoid this problem. Or we can set larger input variation to avoid this problem. In this thesis, we use the second method, because it's less complex. But we have to scarify stability. So, choosing a proper input variation is important.

(ii) Parameters of SISO ESC

There are several parameters in SISO ESC that need to be chosen carefully, which are amplitude of input variation a, frequency of input variation ω, frequency of high-pass filter ω_h.

First, ω is chosen based on the characteristic of the system. ω must be chosen to be faster than the external perturbation but slower than the internal dynamics. Second, amplitude of input variation a should be large enough so that there will be measurable variation in objective function. And a should be large enough to enable the system to go pass low local maxima. Also, larger a results in large converging speed to maxima. However, a cannot be too large, because larger a results in worse stability of the algorithm. Then, ω_h should be chosen based on ω, ω_h must be smaller than ω.

(3) Objective function

Objective function is a value that we defined as a criterion of the performance of the system. In this thesis, we define an objective function for output pulses of ANDi laser. For an ANDi laser, pulses with higher pulse energy and shorter pulse duration is desired. So we define objective function as follow:

$$f(\alpha 3) = \frac{E}{M_4}.$$

Where E is the pulse energy and M_4 is forth moment of the pulse. Forth moment of the pulse is defined as follows:

$$M_4 = \int I(\mathrm{t})(\mathrm{t}-\mathrm{t}_0)^4 dt$$

Where I is intensity, t is time, t_0 is time of pulse peak.

A pulse with shorter pulse duration or pulse power will have a larger objective function. In SISO ESC controlled ANDi laser, we optimize the laser performance by tacking local maxima of objective function.

5. Adaptive controlled ANDi laser

In this thesis, we control the NPR to optimize the performance of ANDi laser. The relation between NPR and pulse quality is complex and nonlinear. SISO ESC is employed to control a HWP in NPR to optimize the output pulse quality.

(1) Schematic for experiment setup

The schematic for experiment setup is shown in Figure.7.

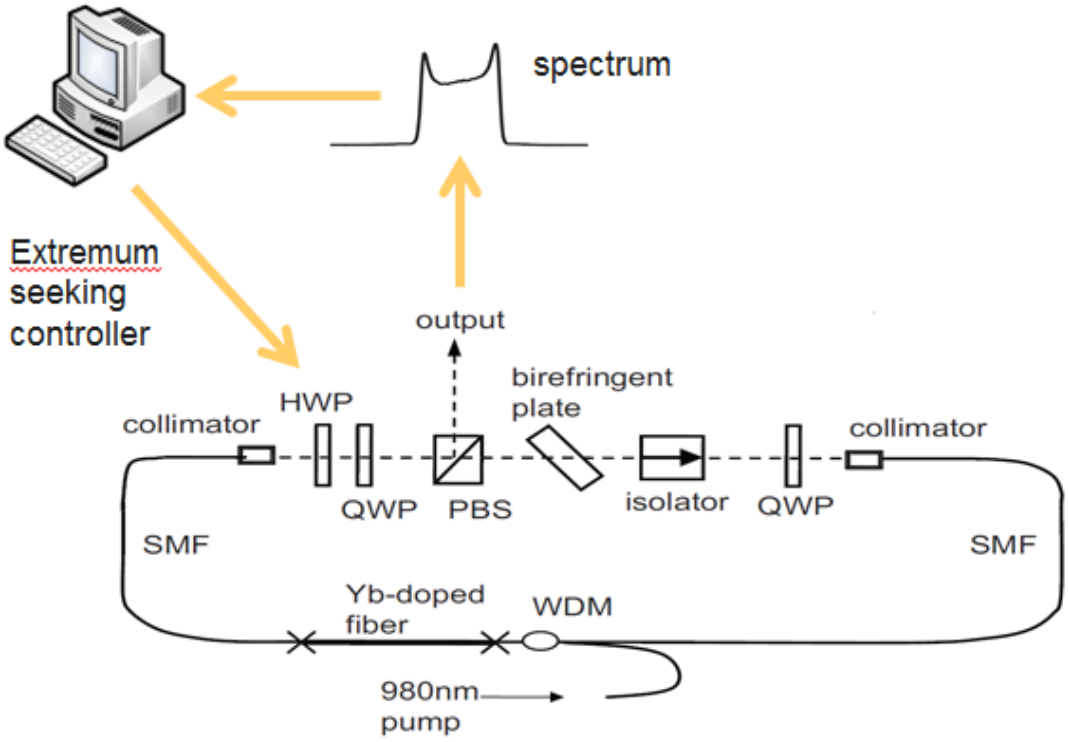

Fig.7 Schematic for experiment set up

The output spectrum is read through a spectrometer. Then the spectrum is analyzed by a PC. And SISO ESC is employed to control the HWP in the cavity.

(2) Relation between angle of HWP and objective function

We rotate HWP from 0 to 180 degree and then rotate back from 180 to 0 degree to investigate the relation between angle of HWP and objective function. The relation between angle of HWP and objective function is shown in Figure.8.

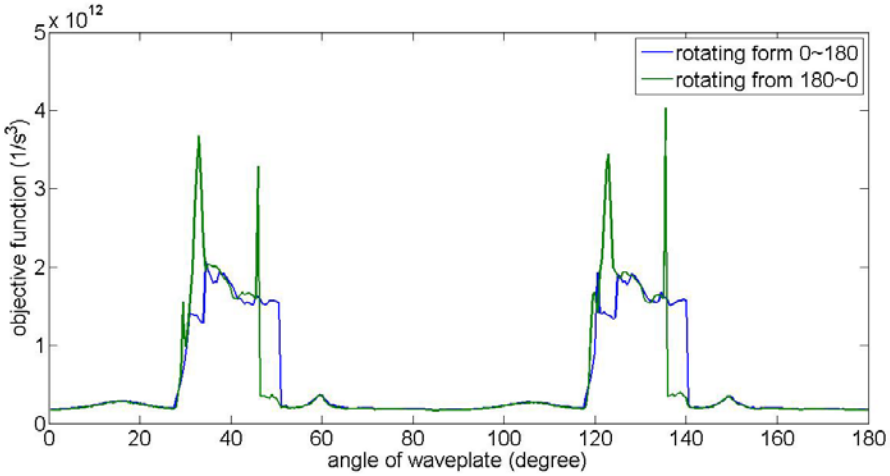

Fig.8 angle of HWP vs objective function

From Figure.8, the relation between angle of HWP and objective function is complex, and analytical solution is difficult to find. Where the value of objective function dramatically increase is the angle of HWP that mode-locks the laser. From the figure, we can conclude that we can mode-lock the laser by rotating HWP. And there are several positions that mode-lock the laser, which indicates that the laser has different mode-locked states. Different mode-locked states lead to different pulse qualities, which can be compared using objective function. Employing SISO ESC, we are able to start from a state which is not mode-locked states to a mode-locked state with largest objective function, in another word, best pulse quality.

One important thing to mention is the hysteresis of ANDi laser[6]. From the figure, we can find that the blue line and the green line don't match each other, which means you sweep the HWP form 0 to 180 degree and record the maxima point, you cannot reach it again by simply rotate the HWP back to the recorded point. The reason is that some mode-locked states are self-start[7], and some mode-locked states are not self-start. The laser cannot directly reach the peaks in the green line. The HWP needs to go over the maxima in the blue line and rotates back to reach the peaks of the green line.

(3) Modification to SISO ESC

SISO ESC works well with adaptive controlled ANDi laser. However, there are several specific problems in this thesis that requires modification to SISO ESC.

First, due to the definition of objective function, it is sensitive to noise far from the center of the pulse, which cannot be avoided during measurement. So a time window needs to be added to the algorithm to ignore noise far from the center of the pulse.

Second, measurement is not so accurate. Inaccurate measurement may cause additional fluctuation to objective function. To avoid the fluctuation, we add an integrator after the time window.

Third, in SISO ESC, previous estimation of the system behavior influence the instant input $\hat{u}$ as shown in Figure.9. As there is a integrator in ESC, memory of previous estimation will cause instability. We multiply the output of the integrator by a value α to avoid instability caused by previous instability. α should be smaller than1. A good estimation of α is 0.9. To further stabilize the laser, we first sweep HWP from 0 to 180 degree, then back to 0 degree. And record the best objective function value. After we start our adaptive control process, PC detects if the laser has reached the

optimized performance. Once the laser reaches the optimized performance, integrator value is set to 0, and SISO ESC is restart.

Taking the three above problems into consideration, SISO ESC is modified as Figure.9.

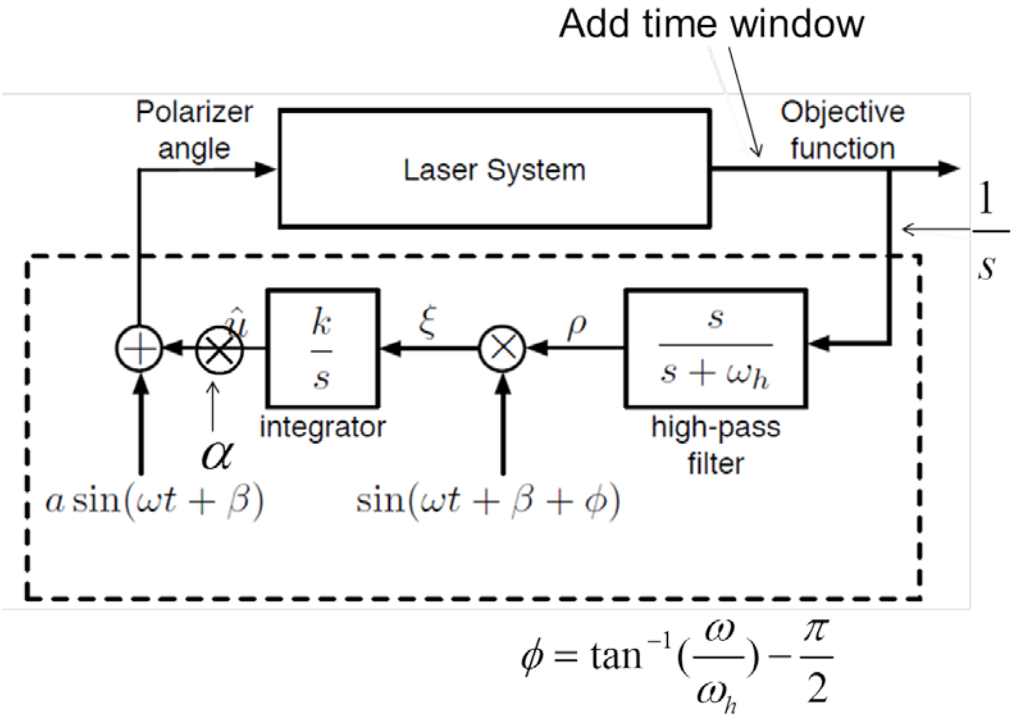

Fig.9 Modified SISO ESC

The modified SISO ESC works well with our ANDi laser. By employing modified SISO ESC, the system starts from a state which is not mode-locked, and end with a stable mode-locked state by adjusting the HWP.

(4) Experiment results

(i) Adaptive control process

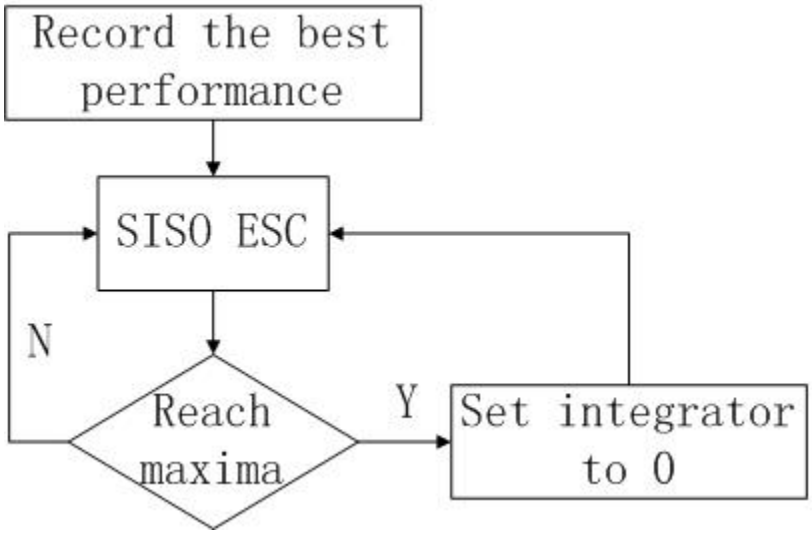

Fig.10 Adaptive control process

As shown in Figure.10. First, the best performance of the laser is recorded by sweeping the angle of HWP. Second, SISO ESC starts. In every round trip of ESC, PC detects if the laser has reached best performance. Once the laser reaches the maxima performance, integrator in ESC is set to 0, then ESC is restart. Set integrator to 0 is to enhance the stability of the algorithm.

(ii) Experiment results

We start the experiment with the laser in a state which is not mode-locked. Then the PC detects the measurement and finds that the laser is not mode-locked. Thus, SISO ESC starts. SISO ESC adjusts the HWP, and the laser finally reaches a optimized state. The result is shown in Figure.10. In Figure.10 (a), normalized objective function starts from 0. With SISO ESC adjusting HWP, objective function increases and finally reaches 1 and stabilizes there. Also, angle of HWP is adjusted from 77 to 88 degree. There is noticeable fluctuation of objective function after the laser is stabled caused by sinusoid input variation. The noticeable fluctuation is because objective function is sensitive. Actually, as we observe the output pulse, there is no noticeable fluctuation in the spectrum.

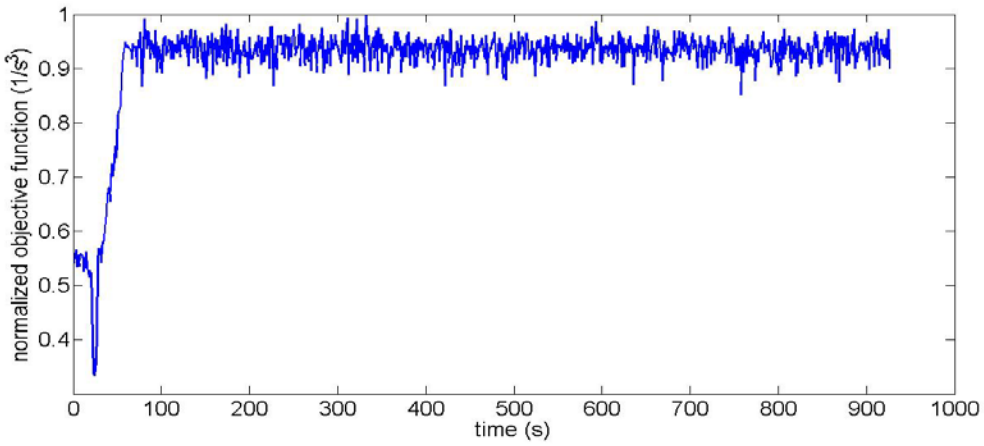

Fig.10 (a) Normalized objective function vs time

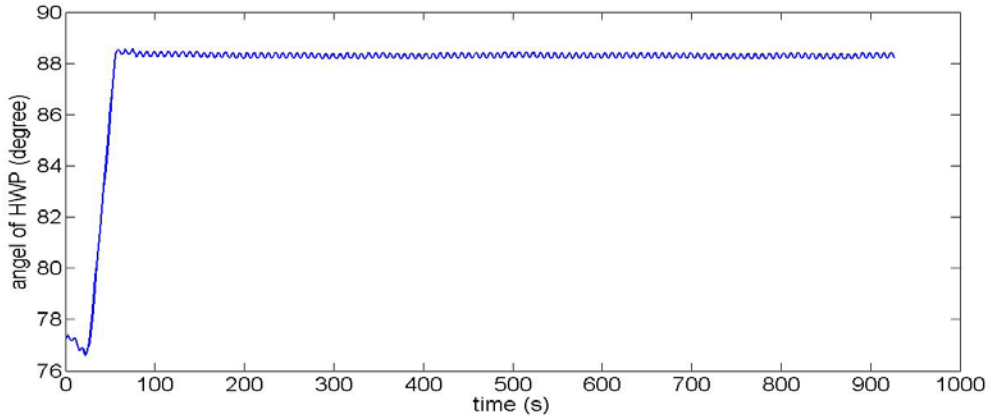

Fig.10 (b) angle of HWP vs time

In reality, some parameters of the cavity may be changed due to environmental variation. For example, birefringence plate is influenced by temperature variation. So we change the angle of birefringence plate after the laser reached the optimized mode-locked state.

As shown in Figure.11, after the birefringence plate is adjusted, the laser loses the mode-locked state. Thus, SISO ESC adjusts the HWP, and finds the optimized mode-locked state again. One important thing to mention, the optimized objective function is larger than the previous mode-locked state. The reason is that SISO ESC optimizes a single input function, so when birefringence plate is adjusted, other parameters in the function is changed. Thus the function is different from the previous one. Because SISO ESC only optimizes one parameter in the system, if all the

parameters need to be optimized, a multi-input-single-output ESC is needed.

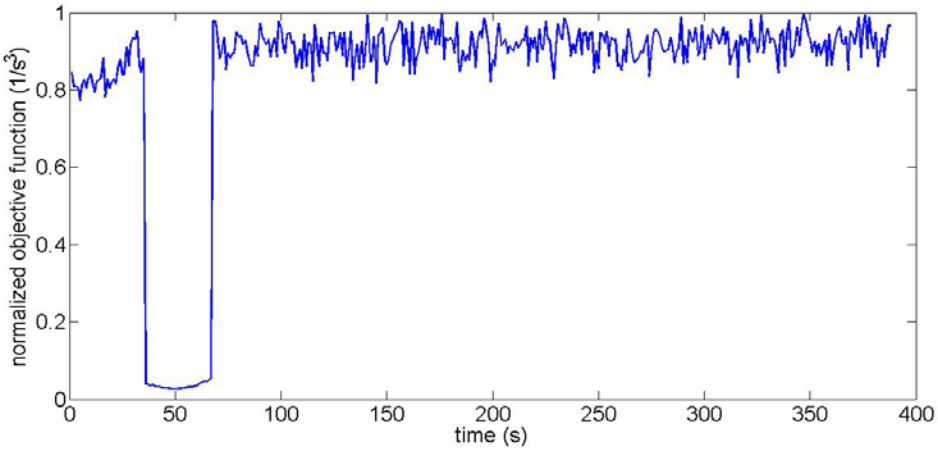

Fig.11 (a) normalized objective function vs time after birefringence plate is adjusted

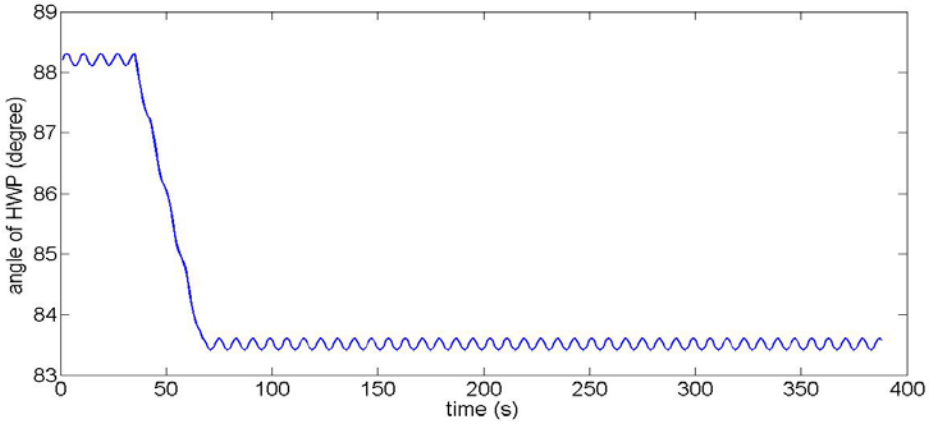

Fig.11 (b) angel of HWP vs time after birefringence plate is adjusted

6. Conclusion

In this thesis, ANDi laser and ESC introduced first. Then we employ SISO ESC to adaptively control ANDi laser. SISO ESC adjusts the angle of HWP to track the optimized performance and stables the ANDi laser. From the experiment results, we can conclude that SISO ESC can find the local maxima performance of ANDi laser and stable the laser when it reached the optimized performance. However, SISO ESC only optimizes one parameter in the cavity which is the angle of HWP. To enhance the

performance of adaptive control, a multi-input-single-output ESC should be employed, but this kind of ESC will be more complex and difficult to realize.

Appendix

(1) Measurement of spectrum

The spectrum of output pulses is measured by spectrometer HR2000. The spectrometer reads the spectrum and the spectrum is analyzed by Matlab. A software package termed omnidriver is needed to use HR2000 with Matlab. It can be downloaded from the following link: http://www.oceanoptics.com/Technical/softwaredownloads.asp

(2) Controlling angle of HWP using Matlab

After the measured spectrum is analyzed by Matlab, SISO ESC controls HWP. We use an electronically controlled rotation stage to control the angle of HWP. The rotation stage we use is PRM1Z8 motorized rotation stage from Thorlabs. PRM1Z8 has a Matlab interface, so the rotation stage can be controlled by Matlab program. The interface comes with a GUI so we can read the angle of HWP, which makes it easy to set the experiment parameters. The GUI is shown in Figure.12.

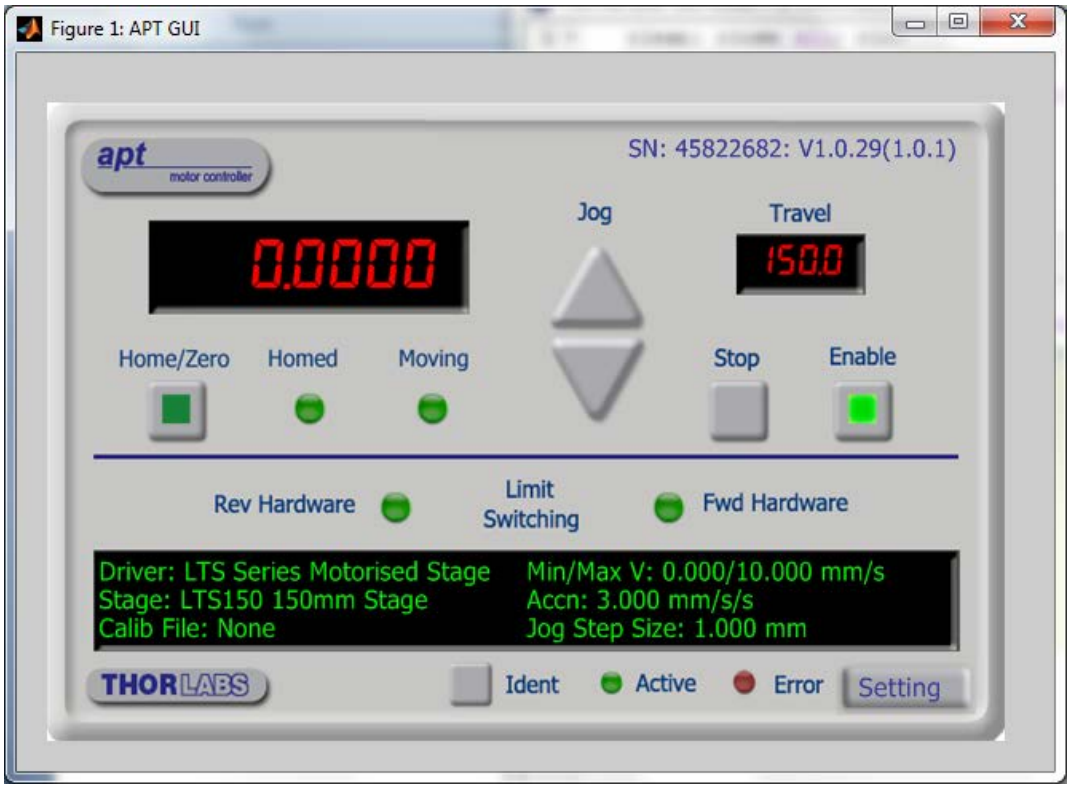

Fig.12 GUI of PRM1Z8

(3) Code

We write a Matlab codes. The first code is to sweep the HWP to record the best performance of ANDi laser. The second code is to adaptively control the HWP by SISO ESC.

(i) Code to record best performance:

```
%% Scan the whole range of the waveplate


%% Create MATLAB Instrument OmniDriver object.
spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');

%% Connect to the instrument.
connect(spectrometerObj);
disp(spectrometerObj);

%% Set parameters for spectrum acquisition.

% integration time for sensor.
integrationTime = 50;
% Spectrometer index to use (first spectrometer by default).
spectrometerIndex = 0;
% Channel index to use (first channel by default).
channelIndex = 0;
```

```
% Enable flag.
enable = 1;

%% Identify the spectrometer connected.

% Get number of spectrometers connected.
numOfSpectrometers = invoke(spectrometerObj,
'getNumberOfSpectrometersFound');

display(['Found ' num2str(numOfSpectrometers) ' Ocean Optics
spectrometer(s).']);

% Get spectrometer name.
spectrometerName = invoke(spectrometerObj, 'getName',
spectrometerIndex);
% Get spectrometer serial number.
spectrometerSerialNumber = invoke(spectrometerObj,
'getSerialNumber', spectrometerIndex);
display(['Model Name : ' spectrometerName])
display(['Model S/N  : ' spectrometerSerialNumber]);

%% Set the parameters for spectrum acquisition.

% Set integration time.
invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
% Enable correct for detector non-linearity.
invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
% Enable correct for electrical dark.
invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);

%% Initialize the motorized rotating stage

global h; % make h a global variable so it can be used outside the
main
          % function. Useful when you do event handling and
sequential           move

%% Create Matlab Figure Container
fpos    = get(0,'DefaultFigurePosition'); % figure default position
fpos(3) = 650; % figure window size;Width
fpos(4) = 450; % Height

f = figure('Position', fpos,...
           'Menu','None',...
           'Name','APT GUI');
%% Create ActiveX Controller
h = actxcontrol('MGMOTOR.MGMotorCtrl.1',[20 20 600 400 ], f);

%% Initialize
% Start Control
h.StartCtrl;
```

```
% Set the Serial Number
SN = 83843499; % put in the serial number of the hardware
set(h,'HWSerialNum', SN);

% Indentify the device
h.Identify;

pause(5); % waiting for the GUI to load up;

%% Controlling the Hardware
%h.MoveHome(0,0); % Home the stage. First 0 is the channel ID
(channel 1)
                 % second 0 is to move immediately



%% Rotate the waveplate and record performance
%Return home before scanning
h.MoveHome(0,0);
%pause;

% 1 degree per rotation
% from 0 to 360

%objective function
object_fct_for = zeros(1,900);
object_fct_back = zeros(1,900);

%wavelengths and frequency
c = 3*10^8;
wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
wavelengths = wavelengths(1410:1710);
frequencies = c./wavelengths;
frequencies =
linspace(min(frequencies),max(frequencies),length(wavelengths));
wavelengths_new = c./frequencies;
time_interval = 2*pi/(max(frequencies)-min(frequencies));
time = 1:length(frequencies);
time = (time-(1+length(frequencies))/2)*time_interval;

h.SetAbsMovePos(0,0);
h.MoveAbsolute(0,1);
pause(10);

for i = 1:900
    % Get the wavelengths of the first spectrometer and save them in
a double array.

    spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);
    spectralData = spectralData(1410:1710); % Intensity
    spectralData_new =
```

```
interp1(wavelengths,sqrt(spectralData),wavelengths_new); %Amplitude
now
    temporalData =
abs(fftshift(ifft(spectralData_new))).^2; %Intensity again
    peak_temp = max(temporalData);
    center = find(temporalData == peak_temp);
    center = center(1);
    Energy = trapz(temporalData);

    M4 = sum(temporalData.*(time-time(center)).^4);
    object_fct_for(i) = Energy/M4;
    h.SetJogStepSize(0,0.2);
    h.MoveJog(0,1);
    pause(2)
    i

figure(2)
plot(wavelengths, spectralData);

    %xlim([1010 1050])
    hold off
    xlim([1010 1050])
    title('Optical Spectrum');
    ylabel('Intensity (counts)');
    xlabel('\lambda (nm)');
    grid on
    %axis tight
drawnow

end

h.SetAbsMovePos(0,180);
h.MoveAbsolute(0,1);
pause(2);

% from 360 to 0
for i = 1:900
     % Get the wavelengths of the first spectrometer and save them
in a double array.
    spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);
    spectralData = spectralData(1410:1710); %Intensity
    spectralData_new =
interp1(wavelengths,sqrt(spectralData),wavelengths_new); %Amplitude
now
    temporalData =
abs(fftshift(ifft(spectralData_new))).^2; %Intensity again
    peak_temp = max(temporalData);
    center = find(temporalData == peak_temp);
    center = center(1);
    Energy = trapz(temporalData);

    M4 = sum(temporalData.*(time-time(center)).^4);
    object_fct_back(i) = Energy/M4;
    h.SetJogStepSize(0,0.2);
```

```
    h.MoveJog(0,2); %opposite direction as rotation1
    pause(2)
    i

figure(2)
plot(wavelengths, spectralData);

    %xlim([1010 1050])
    hold off
    xlim([1010 1050])
    title('Optical Spectrum');
    ylabel('Intensity (counts)');
    xlabel('\lambda (nm)');
    grid on
    %axis tight
drawnow
end

angle1 = linspace(0,180,900);
angle2 = fliplr(angle1);
plot(angle1,object_fct_for,angle1,object_fct_back)
```

(ii) Code for SISO ESC:

```
%% Scan to find the maximum and track the maximum found by scanning

clear all
clc

%% Create MATLAB Instrument OmniDriver object.
spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');

%% Connect to the instrument.
connect(spectrometerObj);
disp(spectrometerObj);

%% Set parameters for spectrum acquisition.

% integration time for sensor.
integrationTime = 50;
% Spectrometer index to use (first spectrometer by default).
spectrometerIndex = 0;
% Channel index to use (first channel by default).
channelIndex = 0;
% Enable flag.
enable = 1;

%% Identify the spectrometer connected.

% Get number of spectrometers connected.
numOfSpectrometers = invoke(spectrometerObj,
'getNumberOfSpectrometersFound');
```

```
display(['Found ' num2str(numOfSpectrometers) ' Ocean Optics
spectrometer(s).']);

% Get spectrometer name.
spectrometerName = invoke(spectrometerObj, 'getName',
spectrometerIndex);
% Get spectrometer serial number.
spectrometerSerialNumber = invoke(spectrometerObj,
'getSerialNumber', spectrometerIndex);
display(['Model Name : ' spectrometerName])
display(['Model S/N  : ' spectrometerSerialNumber]);

%% Set the parameters for spectrum acquisition.

% Set integration time.
invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
% Enable correct for detector non-linearity.
invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
% Enable correct for electrical dark.
invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);

%% Initialize the motorized rotating stage

global h; % make h a global variable so it can be used outside the
main
          % function. Useful when you do event handling and
sequential           move

%% Create Matlab Figure Container
fpos    = get(0,'DefaultFigurePosition'); % figure default position
fpos(3) = 650; % figure window size;Width
fpos(4) = 450; % Height

f = figure('Position', fpos,...
           'Menu','None',...
           'Name','APT GUI');
%% Create ActiveX Controller
h = actxcontrol('MGMOTOR.MGMotorCtrl.1',[20 20 600 400 ], f);

%% Initialize
% Start Control
h.StartCtrl;

% Set the Serial Number
SN = 83843499; % put in the serial number of the hardware
set(h,'HWSerialNum', SN);

% Indentify the device
h.Identify;
```

```
pause(5); % waiting for the GUI to load up;

%% Controlling the Hardware
%h.MoveHome(0,0); % Home the stage. First 0 is the channel ID
(channel 1)
                  % second 0 is to move immediately

%% Algorithm parameter
a = 0.1;
k = 0.02*10^-7;
wh = pi/8;

%objective function
object_fct = zeros(1,2);
object_fct_filted = zeros(1,2);
integrate_value = 0;
Maxintegrate_value = 0;
n = 0;

%wavelengths and frequency
c = 3*10^8;
wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
wavelengths = wavelengths(1410:1710);
frequencies = c./wavelengths;
frequencies =
linspace(min(frequencies),max(frequencies),length(wavelengths));
wavelengths_new = c./frequencies;
time_interval = 2*pi/(max(frequencies)-min(frequencies));
time = 1:length(frequencies);
time = (time-(1+length(frequencies))/2)*time_interval;
time_window = zeros(1,length(frequencies));
time_window(floor(9*length(frequencies)/20):floor(11*length(frequenc
ies)/20)) = 1;

%first acquizition

%get objective function
spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);
spectralData = spectralData(1410:1710); %Intensity
spectralData_new =
interp1(wavelengths,sqrt(spectralData),wavelengths_new); %Amplitude
now
temporalData = abs(fftshift(ifft(spectralData_new))).^2; %Intensity
again
temporalData = temporalData.*time_window;
peak_temp = max(temporalData);
center = find(temporalData == peak_temp);
center = center(1);
Energy = trapz(temporalData);
M4 = sum(temporalData.*(time-time(center)).^4);
object_fct(1) = Energy/M4;
m = 1;
%record objective function
```

```
object_record(m) = object_fct(1);

%record position of waveplate
Position_record(m) = h.GetPosition_Position(0);



%% Searching for mode locking position and stablize the mode
while (1)
    %input sin signal
    n = n+1;
    if n == 8
        n = 0;
    end
    Step_size = a*(sin(pi*n/4)-sin(pi*(n-1)/4));
    if Step_size > 0
        h.SetJogStepSize(0,Step_size);
        h.MoveJog(0,1);
    else
        Step_size = -Step_size;
        h.SetJogStepSize(0,Step_size);
        h.MoveJog(0,2);
    end

    pause(0.5)
    spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);
    spectralData = spectralData(1410:1710); %Intensity
    spectralData_new =
interp1(wavelengths,sqrt(spectralData),wavelengths_new); %Amplitude
now
    temporalData =
abs(fftshift(ifft(spectralData_new))).^2; %Intensity again
    temporalData = temporalData.*time_window;
    peak_temp = max(temporalData);
    center = find(temporalData == peak_temp);
    center = center(1);
    Energy = trapz(temporalData);
    M4 = sum(temporalData.*(time-time(center)).^4);
    object_fct(2) = Energy/M4;

    %record
    m = m+1;
    object_record(m) = object_fct(2);
    %filter
    object_fct_filted(2) = (object_fct(2)-
object_fct(1)+object_fct_filted(1))/(1+wh);
    %integrate
    integrate_value =
integrate_value+object_fct_filted(2)*k*sin(pi*n/4+pi/3);
    if integrate_value > 0
        h.SetJogStepSize(0,integrate_value);
        h.MoveJog(0,1);
    else
        h.SetJogStepSize(0,-integrate_value);
```

```
        h.MoveJog(0,2);
    end

    %record max integrate value
    if abs(integrate_value) > Maxintegrate_value
        Maxintegrate_value = abs(integrate_value);
    end
    pause(0.5)
    object_fct(1) = object_fct(2);
    Position_record(m) = h.GetPosition_Position(0);

%% Aquire the spectrum.
%wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
% Get the wavelengths of the first spectrometer and save them in a
double
% array.
%spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);

figure(2)
plot(wavelengths, spectralData);

    %xlim([1010 1050])
    hold off
    xlim([1010 1050])
    title('Optical Spectrum');
    ylabel('Intensity (counts)');
    xlabel('\lambda (nm)');
    grid on
    %axis tight
drawnow

end
plot(object_record)
```

References

[1] Frank W. Wise, Andy Chong, and William H. Renninger, Laser&Photon.Rev. **2**, 58 (2008).

[2] H. A. Haus, E. P. Ippen, and K. Tamura, IEEE Journal of Quantum Electronics **30**, 200 (1994).

[3] Xuling Shen, Wenxue Li, Ming Yan and Heping Zeng, Optics Letters **37**, 3426 (2012).

[4] Steven L. Brunton, Xing Fu, and J. Nathan Kutz, IEEE Journal of Quantum Electronics **49**, 852 (2013).

[5] Andy Chong, William H. Renninger and Frank W. Wise, J. Opt. Soc. Am. B **25**, 140 (2008).

[6]Edwin Ding, William H. Renninger, Frank W. Wise, Philippe Grelu, Eli Shlizerman and J. Nathan Kutz, International Journal of Optics, Vol.2012, Article ID 354156.

[7] Ariel Gordon1, Omri Gat, Baruch Fischer and Franz X. K¨artner, Optics Express **14**, 11142 (2006).